AD A042003

AD No.
DDC FILE COPY

Naval Warfare Research Center

Technical Note
NWRC-TN-37

April 1972

SIMULATION MODELS OF SEARCH IN THE PRESENCE OF DECOYS

By: E. L. WONG

Prepared for:

NAVAL ANALYSIS PROGRAMS (Code 462)
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-71-C-0119
Task No. NR 274-008-34

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

TECHNICAL REPORT

A Technical Report is a document of Stanford Research Institute that presents results of work directed toward specific research objectives. The report is a comprehensive treatment of the objectives, scope, methodology, data, analyses, and conclusions, and presents the background, practical significance, and technical information required for a complete and full understanding of the research activity. Technical Reports are reviewed and approved by a division executive director or higher official of the Institute.

RESEARCH MEMORANDUM

A Research Memorandum is a document of Stanford Research Institute that presents the results of work in progress. The purpose of the Research Memorandum is to invite comment on research in progress. It is a comprehensive treatment of a single research area or of a facet of a research area within a larger field of study. The Memorandum presents the background, objectives, scope, summary, and conclusions, as well as method and approach, in a condensed form. The report is reviewed and approved by a department director or higher official of the Institute.

TECHNICAL NOTE

A Technical Note is a working paper that presents the results of research related to a single phase or factor of a research problem. The purpose of the Technical Note is to instigate discussion and criticism. It presents the concepts, findings, and/or conclusions of the author. The report is reviewed by a project leader or higher official of the Institute.

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

Naval Warfare Research Center *April 1972*

Technical Note
NWRC-TN-37

SIMULATION MODELS OF SEARCH IN THE PRESENCE OF DECOYS

By: E. L. WONG

Prepared for:

NAVAL ANALYSIS PROGRAMS (Code 462)
OFFICE OF NAVAL RESEARCH
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-71-C-0119
Task No. NR 274-008-34

SRI Project 1016-245

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute Menlo Park, California 94025	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP --

3. REPORT TITLE

Simulation Models of Search in the Presence of Decoys

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Note

5. AUTHOR(S) (First name, middle initial, last name)

Ellen L. Wong

6. REPORT DATE April 1972	7a. TOTAL NO. OF PAGES 74	7b. NO. OF REFS 1

8a. CONTRACT OR GRANT NO. N00014-71-C-0119 b. PROJECT NO. RF 018-02-06 c. NR 274-008-34 d.	9a. ORIGINATOR'S REPORT NUMBER(S) NWRC-TN-37 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Analysis Programs (Code 462) Office of Naval Research Arlington, Virginia 22217

13. ABSTRACT

A simulation model that represents a submarine's search for a high value target within a specified operation has been developed. The model, constructed as an adjunct to the formulation and implementation of a computationally more efficient analytical model, supported the assessment of the potentials of tactical deception techniques in anti-submarine warfare. This technical note describes the details of the model structure. The assessment results are published in a separate, classified, final project report.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED
Security Classification

253000

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Modeling						
Simulation						
Submarine search process						
Antisubmarine warfare						
Decoys						
Submarine motion						
Semi-Markov models of search						
Submarine search for a HVU						

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

PREFACE

The work reported in this technical note was conducted as a subtask within a larger project directed toward the assessment of tactical deception in antisubmarine warfare. The project was sponsored by Naval Analysis Programs, Mr. R. J. Miller, Director, in the Office of Naval Research. Mr. J. G. Smith was the ONR Project Scientific Officer.

The research effort was performed by the Naval Warfare Research Center, Mr. L. J. Low, Director, of Stanford Research Institute. Mr. A. Bien of NWRC was the project leader.

The author wishes to thank Mr. M. W. Zumwalt for his invaluable help in the preparation of the simulation model description.

CONTENTS

ILLUSTRATIONS

TABLES

1. INTRODUCTION

A simulation model that represents a submarine's search for a high value target within a specified operating area is described in this report. This model was developed as an adjunct to the formulation and implementation of a computationally more efficient analytical model.*
The simulation model served two purposes. First, the simulation model provided a validation of the statistical inputs used for the analytical model. Specifically, the simulation studies validated the applicability of the analytical model for determining rate of encounter between submarine and targets. Second, results obtained through exercise of the simulation model provided a convenient check of the reasonableness of analytical model results.

---

* J. M. Moore; "Semi-Markov Models of Search in the Presence of Decoys," NWRC RM-64, SRI Project 3016-245, Contract N00014-71-C-0119; Stanford Research Institute, Menlo Park, California; July 1971 (UNCLASSIFIED)

2. GENERAL SIMULATION MODEL CHARACTERISTICS

The overall simulation model is comprised of three separate computer submodels. Each of these submodels is designed to represent a specific set of operational conditions which may characterize the various phases of submarine search for a high value target (HVT) in a specified area. The operational conditions represented in these submodels include continuous submarine search--with and without false targets (decoys) present, and submarine search employing the sprint-drift tactic without false targets present.

Each of these submodels is programmed in the BASIC algebraic language for processing and operation by a time-share operating system incorporating a CDC-6400 computer. Because of time-share system limitations, particularly with regard to output of results, these submodels are constructed so as to facilitate independent operations.

Those model characteristics which are common to all three submodels are described and discussed in this section; whereas pecularities of each submodel are discussed in the sections which follow.

The primary output of each submodel is the elapsed time to HVT detection by the submarine searcher and various statistical quantities which characterize the distribution of the time to HVT detection. Specific

aspects of the various common model characteristics are discussed in the paragraphs which follow.

2.1 Operational Framework

In every case, the area is considered to be circular in shape with a known radius, R. While the radius of the area may be inputted as a parameter, it has usually been assigned a value of 200 nmi.

For reference purposes, the center of the area is made to coincide with the origin of a cartesian coordinate system, oriented so as to position the X-axis in a horizontal direction. Initial and subsequent participant positions are expressed in terms of the underlying coordinate system.

The initial positions of the HVT and decoys (if included) are randomly distributed within the area. This is accomplished through the use of random numbers uniformly distributed on the interval [0,1] in the following manner.

Let u_1 and u_2 be two numbers drawn randomly from a population of numbers which is uniformly distributed on the interval [0,1]. Then, for example, the initial position of the HVT, P_{co}, is determined as:

$$P_{co} = (Ru_2, 2\pi u_1) = [Ru_2 \cos (2\pi u_1), Ru_2 \sin (2\pi u_1)] .$$

Initial positions for the decoys are determined in a similar manner utilizing, of course, different random numbers.

In the case of the submarine searcher, the model user may exercise an option to select either an initial position inside the area or a position on the boundary of the area. In this latter case, the initial submarine position, P_{so}, would be (u_3 is again a random number):

$$P_{so} = (R, 2\pi u_3) = [R \cos (2\pi u_3), R \sin (2\pi u_3)] \quad .$$

In the former case, the procedure would be similar to that described above for determining the initial position of the HVT.

For the majority of the cases investigated, an initial submarine position on the boundary of the search area was selected. This condition was considered to more accurately describe the actual operational situations which might be encountered.

2.2 Model Time

Each of the submodels employs a fixed time-step form of operation to record elapsed program time. Thus, participant motion and the occurrence/non-occurrence of various events are evaluated at constant, discrete points in time. This form of model construction necessitates a tradeoff between the size of the time-step and the concurrent actual running time of the program. Currently, a time-step of 0.5 hours is used. This interval provides a sufficiently detailed description of the interactions of the various forces, while not requiring exorbitant program running time. Of course, the size of the time-step may be inputted prior to program utilization.

2.3 Submarine/HVT/Decoy Motion

Once the initial positions of the various units are determined, the motion of each is the result of a pseudo-random process. That is, the choice of direction is purely random, but the unit speed and length of time on each leg are predetermined. Further, because of the finite nature of the boundary of the area as well as other operational considerations, the motion of the various units may be inhibited to conform to these conditions. Area boundary effects are discussed in this section; other operational factors which inhibit unit motion are discussed in the appropriate sections which follow.

A typical search initiation situation is illustrated in Figure 2-1. Where, as previously described,

$$P_{co} = (Ru_2, 2\pi u_1)$$

and

$$P_{so} = (R, 2\pi u_3) \quad .$$

The initial direction of HVT motion is then the angle $\theta_{co} = 2\pi u_4$, that is, the initial HVT movement is radially away from the center of the area. This movement is indicated by the dashed line originating from P_{co} in Figure 2-1. The searcher, on the other hand, moves initially toward the center of the area in a direction equal to $\theta_{so} = 2\pi u_5$. It should be noted that the angles θ_{co} and θ_{so} are measured counterclockwise from the positive X-axis and do not, therefore, correspond to the usual East West, North-South orientation. Decoys, if not stationary, move initially in a manner similar to that exhibited by the HVT. No further

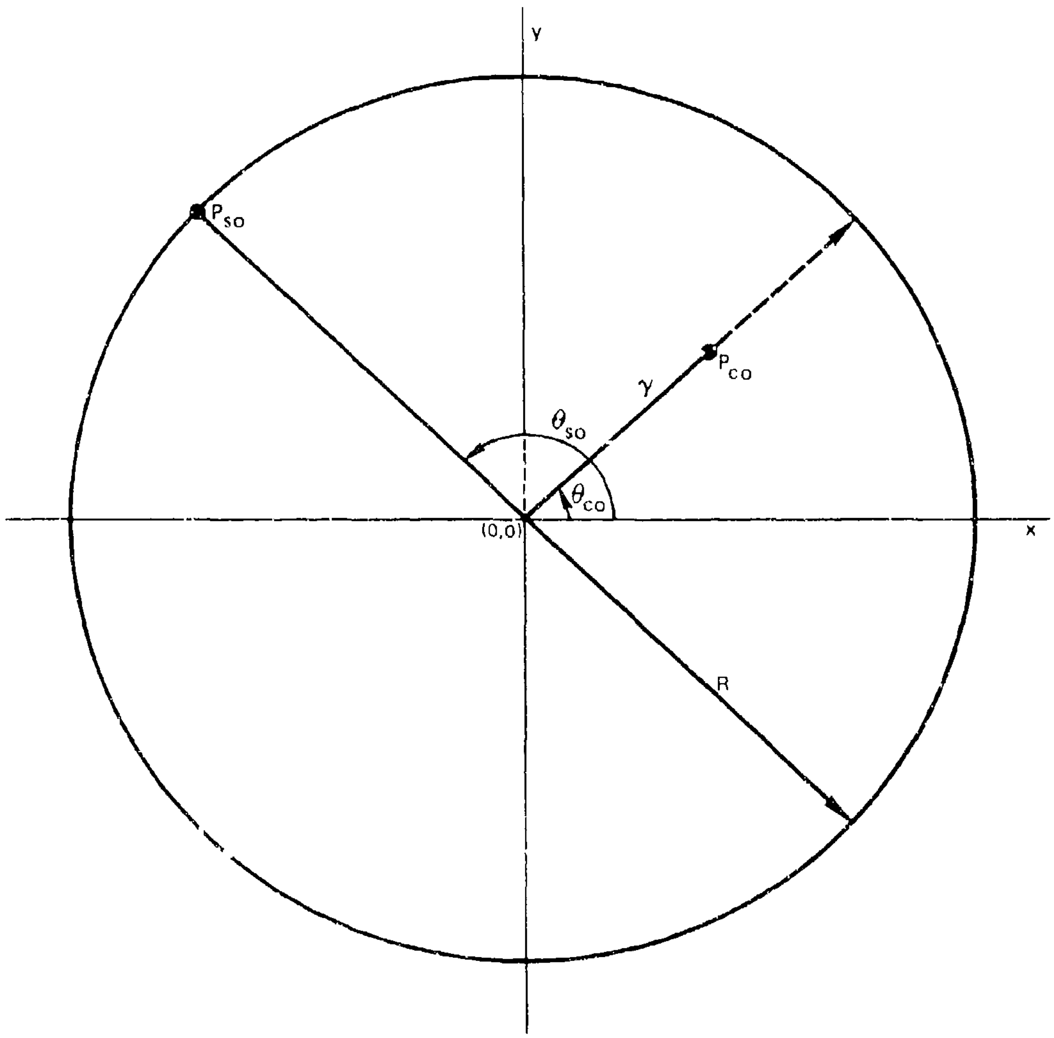

FIGURE 2-1 TYPICAL SEARCH INITIATION SITUATION

comment will be made in this section regarding subsequent decoy movement, for this aspect of the model formulation will be discussed in the section dealing with the peculiarities associated with search in the decoy field environment.

Control over the movement of the HVT and the submarine is exercised in two ways: by specifying, as a program input, the length of each track segment; and automatically, by "reflecting" either the HVT or the submarine off the boundary of the area.

As an illustration of these two procedures, consider that the length of the HVT track segments has been specified (inputted) as a maximum of L_c nmi. Then, using the situation depicted in the previous diagram, if $\gamma + L_c \leq R$, the HVT will move along the dashed line until either a distance L_c from P_{co} is reached or the HVT is detected by the submarine.

For either condition: HVT completes movement of L_c nmi or HVT at the boundary of the area, a new direction of movement is determined by calculating a "turning angle," θ', such that

$$\theta' = 2\pi u \qquad \text{(u a random number)}$$

and the new direction of movement is then

$$\theta_{new} = \theta_{old} + \theta'$$

subject to the constraint that, if the HVT is outside the area, θ_{new} must result in returning the HVT to a posi ion within the area.

It should be apparent from the foregoing discussion that the speed of the HVT (and the submarine) is specified prior to beginning the model run and is constant throughout the run.

Movement of the submarine is accomplished in a manner exactly similar to that described for the HVT.

It should be noted that the effect of keeping the submarine within the area boundary reflects an implicit assumption that the submarine possesses perfect information concerning the size and location of the search area. If this were not the case, it would be necessary to overlay the circular area with an area representative of the degree of submarine intelligence information. Such a submarine area might be larger, smaller, and/or offset from the actual area.

2.4 Detection of the HVT and/or Decoy

In each of the submodels, the detection capability of the submarine against either the HVT or the decoy is described by a definite range probability law or "cookie cutter." That is, for some range R_D, the probability of detection is

$$p\,[\mathrm{Det}] = \begin{cases} 1 & r \leq R_D \\ 0 & r > R_D \end{cases}$$

where r is the range between the submarine and either the HVT or the decoy at the end of a particular time-step. In this regard, it is possible for the submarine to pass the HVT or decoy within detection range during

a time-step without a detection being recorded if the range between the final positions is sufficiently large. Again, judicious selection of the time-step, keeping the possible relative speeds in mind, can serve to minimize this possibility.

In actual operation, a value of R_D may be specified for both the HVT and the decoy which reflects characteristics of the HVT and decoy, the various environmental effects, and the other factors which determine detectability of HVT and decoy. Obviously, one method of reflecting the efficiency of the decoy in accurately impersonating the HVT is to make the detection range of the decoy approach that of the HVT.

2.5 Input Parameters

In a slight departure from the previous format of presenting only the common features of the various submodels, all of the necessary input parameters for each of the submodels are presented in Table 2-1. The entries in Table 2-1 are so arranged as to present the common inputs required and then the specific inputs required for each submodel. For ease of reference, the BASIC parameter symbology for the various input quantities is indicated in Table 2-1 rather than the more traditional mathematical notations.

Because of the random aspects introduced into the models, the results from each of the submodels are amenable only to statistical interpretation. For this reason, it is necessary to replicate the model results many times

Table 2-1

SIMULATION MODEL INPUT PARAMETERS

Basic Symbol	Definition	Units
	<u>Common Parameters</u>	
VC	HVT Velocity	kt
COURC	Length of HVT Track Segment (COURC ≤ R)	nmi
R	Radius of Objective Area	nmi
DT	Time Step Increment	hr
NREPLI	Number of Program Replications (Note: NREPLI ≤ 500 for Continuous Search--With False Targets Submodel)	Integer
	<u>Continuous Search--Without False Targets</u>	
RO	Radius of Detection of HVT by Submarine	nmi
VS	Submarine Search Velocity	kt
COURS	Length of Submarine Track Segment (COURS ≤ R)	nmi
	<u>Sprint/Drift Search--Without False Targets</u>	
VSSP	Submarine Sprint Velocity (VSSP·SPP ≤ R)	kt
VSDF	Submarine Drift Velocity (VSDF·DFP ≤ R)	kt
SPP	Submarine Sprint Period	hr
DFP	Submarine Drift Period	hr
RDS	Radius of Detection of HVT by Submarine During Sprint	nmi
RDD	Radius of Detection of HVT by Submarine During Drift	nmi
LFILE LREC IP	Program Restart Parameters--for Use in Case of Program Execution Interruption. Initial Values: LFILE = 1, LREC = 0, IP = 1	Integer Integer Integer
	<u>Continuous Search--With False Target Field</u>	
VS	Submarine Search Velocity	kt
COURS	Length of Submarine Track Segment (COURS ≤ R)	nmi
COURD(I)	Length of Decoy Track Segment [I = 1,2,3,4,5] [COURD(I) ≤ R]	nmi
VDC(I)	Decoy Velocity [I = 1,2,3,4,5] [VDC(I) > 0]	kt
TC	Decoy Classification Time	hr
TM	Decoy "Turned Off" Time	hr
R1R	Radius of Detection of Decoy by Submarine	nmi
R2R	Radius of Detection of HVT by Submarine	nmi

for each set of operational conditions which are to be investigated. Thus, provision is included in each submodel for user specification of the number of replications of model results desired. It should be noted that, because of the increased computational time due to the added presence of decoy, the number of replications for the continuous search with the decoy submodel must be limited to less than, or at most, 500.

All other entries in Table 2-1 are either self-explanatory or are discussed elsewhere in this technical note.

2.6 Submodel Outputs

In each case, the principal quantity measured is the elapsed time to first detection of the HVT by the submarine. Since detection is characterized by a definite range probability law, first detection is equivalent to first encounter, where encounter occurs whenever the range between the submarine and the HVT is less than some predetermined value.

The output of each submodel consists then of the sample mean, variance, and standard deviation of the time to first detection. For the two continuous search submodels, these values are indicated by the BASIC symbols: MEANC, VARC, and STDEVC, respectively. For the sprint drift submodel, the output format has been modified to print out the full titles: "STANDARD DEVIATION," etc.

In computing the sample variance, the consistent estimator formulation for the population variance is used, that is:

$$S^2 = \frac{1}{M} \sum_{i=1}^{M} (T_i - \bar{T})^2 \quad ,$$

where $\bar{T}$ is the sample mean $\left(= \frac{1}{M} \sum_{i=1}^{M} T_i \right)$ and S^2 is the sample variance.

The output of the two continuous search submodels also includes both a frequency count and a cumulative frequency of occurrence of first detection as a function of elapsed time interval. Further, the output of the continuous search with the decoy submodel includes a listing of specific times of first encounter of the HVT with the submarine for each of the replications.

3. CONTINUOUS SEARCH - NO DECOY SUBMODEL

This submodel represents the basic structure for all of the submodels constructed to date. As such, the internal structure of this submodel is exactly similar to that described in the preceding section. A single submarine searcher seeks a single HVT within a delineated objective operating area. The two units move in a pseudo-random manner, staying within the objective area at all times. Each case or replication is terminated at the instant of initial submarine-HVT encounter/detection. The results obtained from exercising this submodel for ranges of the various input parameters provide the baseline data for evaluating the effectiveness of various tactical procedures, ACM employment policies, and combinations of both.

A typical example of the type of motion which this submodel generates, records, and evaluates is shown in Fig. 3-1.

An example of the results obtained from exercising this submodel is presented in Fig. 3-2.

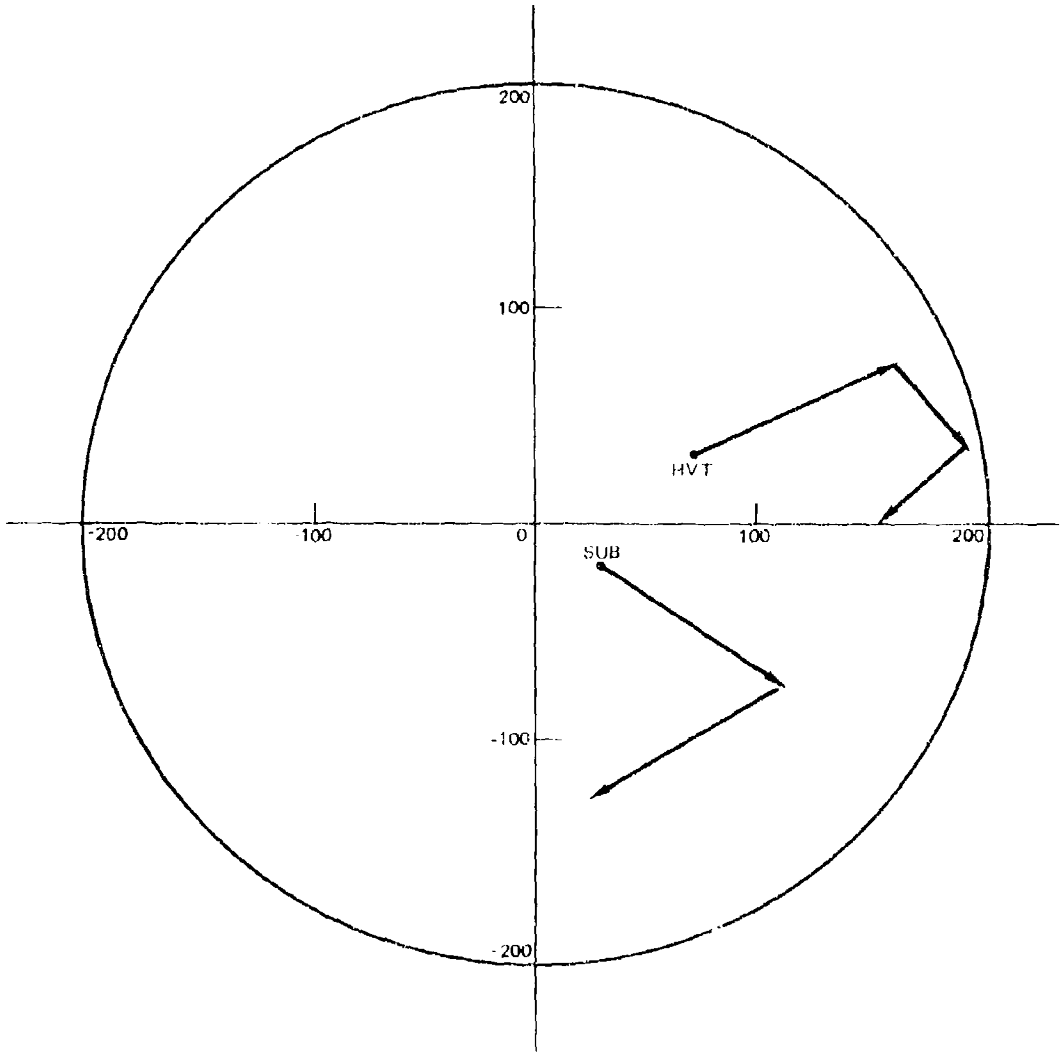

FIGURE 3-1 SAMPLE PLOT OF HVT AND SUBMARINE MOTION

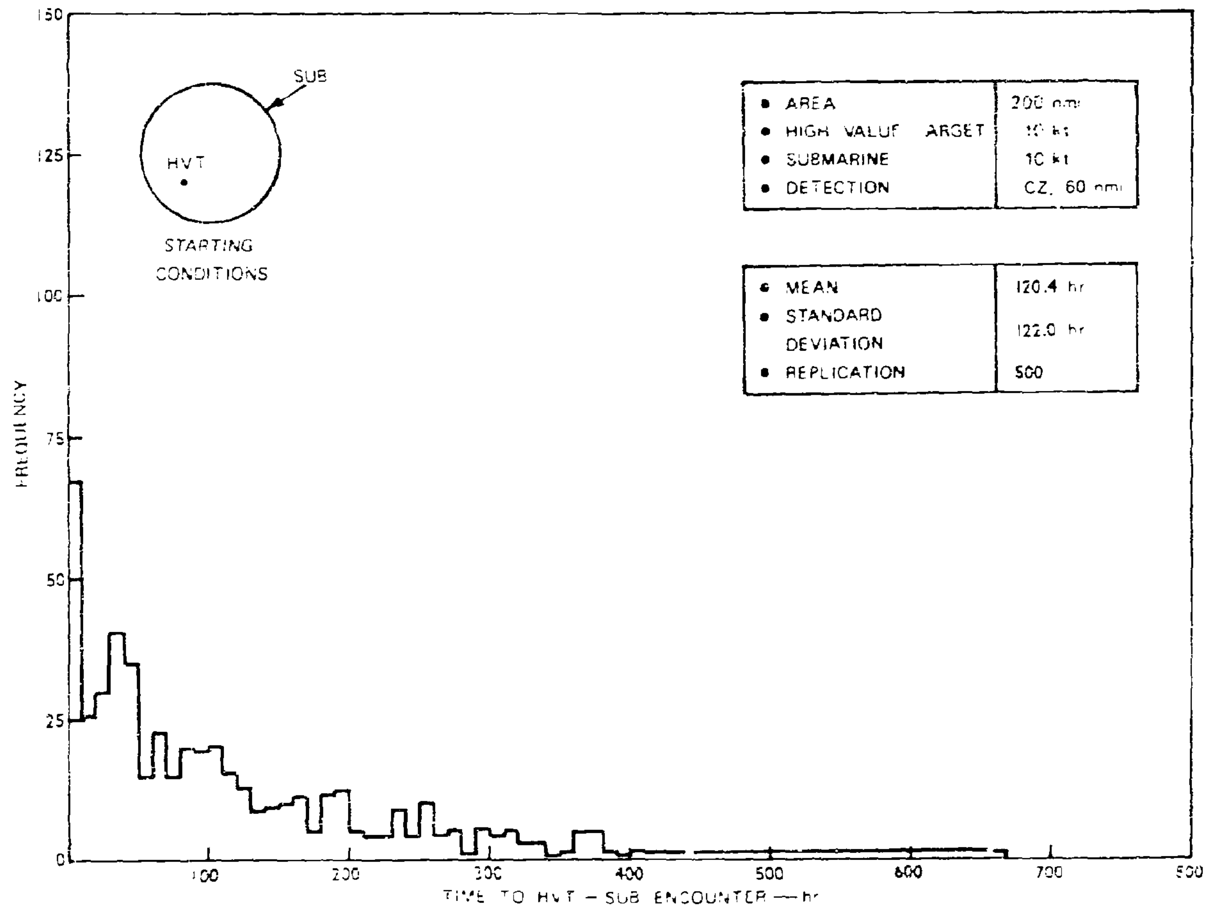

FIGURE 3-2 SAMPLE RESULTS FROM THE CONTINUOUS SEARCH — NO DECOY SUBMODEL

4. SPRINT/DRIFT SEARCH - NO DECOY SUBMODEL

This submodel differs from the baseline submodel in two principal aspects: the manner in which the motion of the submarine is regulated and the manner in which detection of the HVT may occur.

As indicated in the title of this submodel, the submarine employs a tactic of first executing a high speed sprint and then a slow speed drift to attempt detection of the HVT. Thus, motion of the submarine is characterized by four input parameters: the sprint speed, the sprint period, the drift speed, and the drift period. In this manner, the single length of submarine track segment (COURS) specified in the baseline submodel is replaced by two track lengths equal to (VSSP·SSP) and (VSDF·DFP), respectively. (See Table 2-1 for definition of symbols.) At the end of each of these track lengths, new, random headings are determined for the submarine. This procedure is illustrated in the example plot of submarine and HVT movement presented in Figure 4-1.

Under actual conditions, a submarine is effectively "acoustically blind" during the sprint period. There may be instances, however, when the speed selected for the sprint tactic is not so high as to completely eliminate the detection capability of the submarine, especially if the submarine and the HVT should pass close aboard one another. For this reason, provision is included in this submodel for specifying a submarine

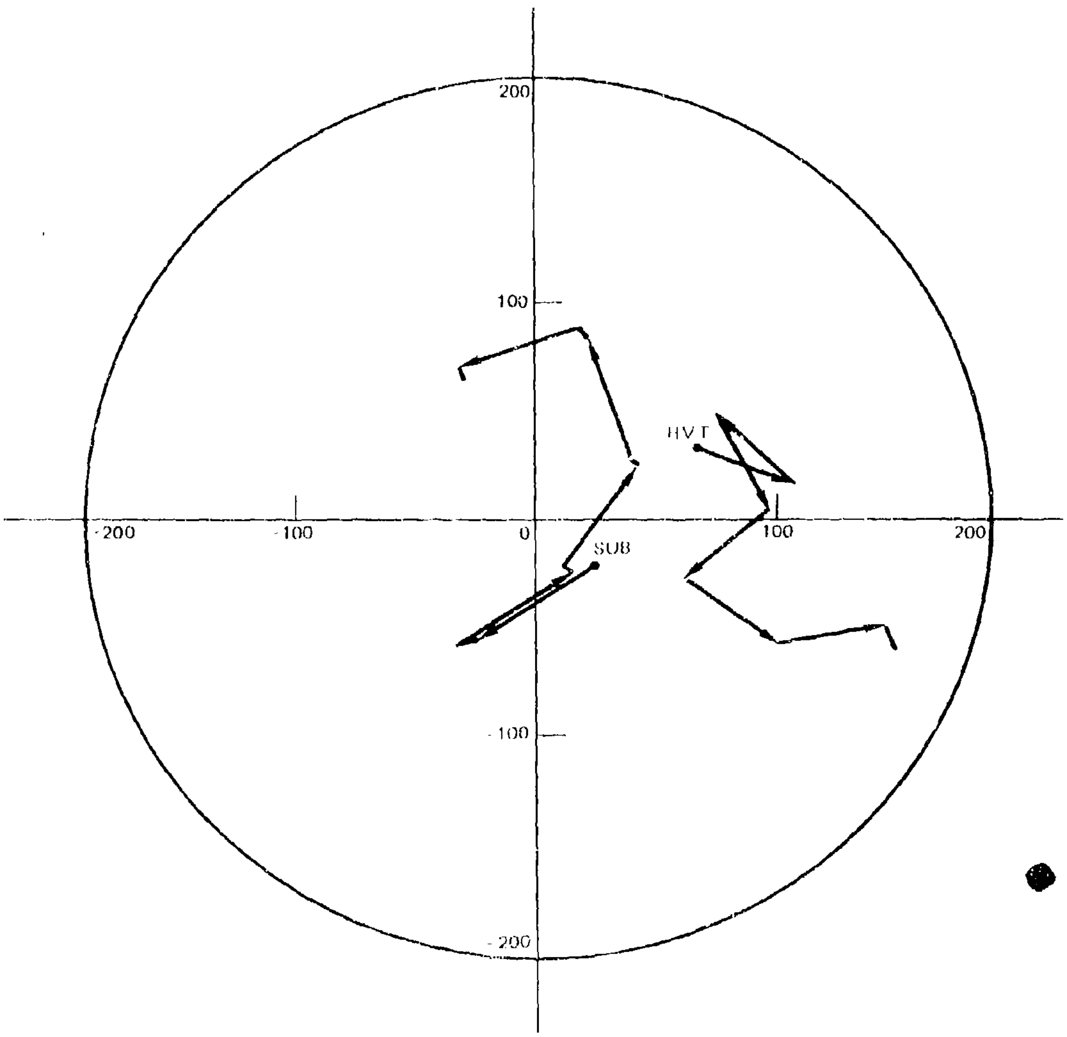

FIGURE 4.1 SAMPLE PLOT OF MOTIONS OF HVT AND SUBMARINE IN THE SPRINT/DRIFT MODEL

detection capability during the sprint period although, in all runs to date, this value has been set so low as to effectively preclude such detections.

Four programs are required to exercise this submodel. These programs are: "SPRINT," "FILECR," "UPDATE," and "STATICS." The last of these, "STATICS," performs the statistical analysis of the results of the various replications and prints the summary data.

The program, "SPRINT," comprises the main program of this submodel and models all of the details of the search. As such, this program requires specification of all of the initial input parameter values indicated in Table 2-1. Prior to the execution of SPRINT, an independent subroutine, LITBAL, must be called into the processor. This is accomplished by the system command, "Get, LITBAL."

The two programs, "FILECR" and "UPDATE," are utility programs that manipulate permanent program data files. These programs were found to be necessary in order to safeguard against loss of accumulated data in the case of machine failure or program interruption during lengthy program executions. The three parameters, "LFILE," "LREC," and "IP" are used to recover and restart program execution should it be interrupted. (LFILE is the BASIC symbol for "Last File Updated," and LREC is the symbol for "Last Record Created.")

The output from the SPRINT program is printed on magnetic tape after every 20th replication. This output is then available for either

statistical analysis, using the STATIC program, and/or permanent storage on magnetic tape using the UPDATE program.

An example of the results obtained from exercising this submodel is presented in Figure 4-2.

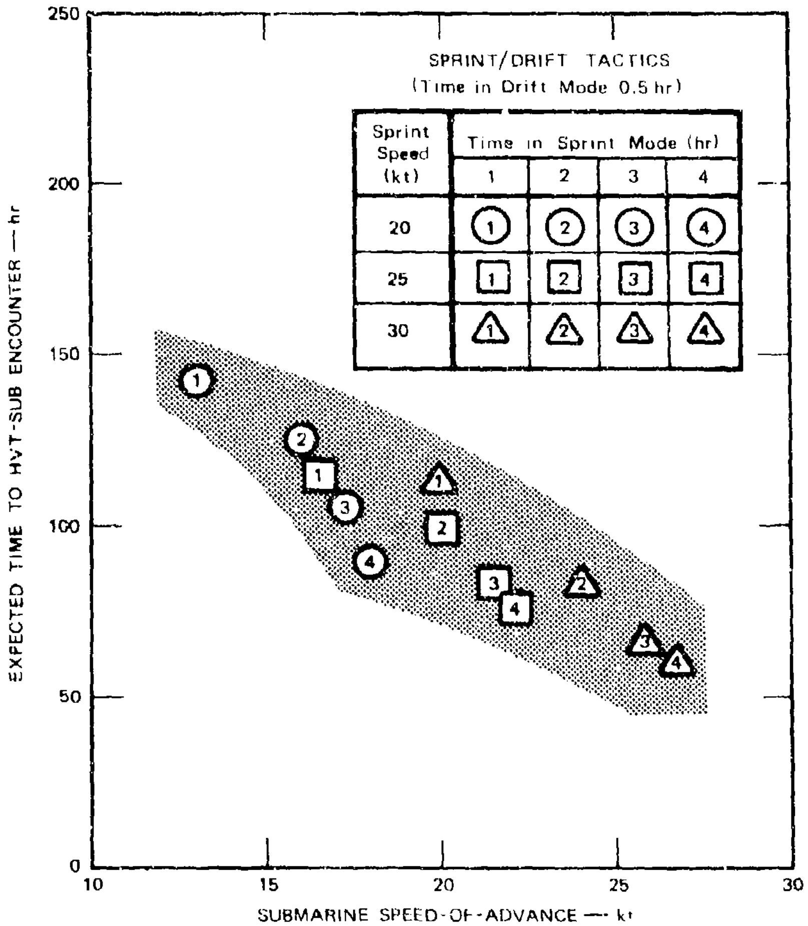

FIGURE 4-2 SAMPLE RESULTS FROM THE SPRINT/DRIFT SEARCH — NO DECOY SUBMODEL

5. CONTINUOUS SEARCH - DECOY FIELD SUBMODEL

As the title implies, this submodel incorporates the effects on searcher effectiveness due to the presence of decoys within a specified area. This innovation requires several departures from the procedures established for the baseline submodel.

As before, the submodel includes a single submarine searcher and a single HVT. Due to limitations on computer core storage in the present time-share system, the number of decoys must be restricted to five or less. Increased capacity can be achieved through the use of utility programs such as those described in the preceding section, but this has not been done at this time.

Either stationary or moving decoys can be simulated through appropriate specification of the decoy velocity, VDC(I). Because of some of the computations involving VDC, however, this value cannot be zero. Therefore, the stationary decoy is represented by making VDC very small, albeit, positive.

The HVT and the decoys are initially positioned within the area; the submarine searcher on the boundary. All of the units are permitted to move in the pseudo-random manner described before subject to the provision that no decoy may approach the HVT closer than (R1R + R2R)nmi. (See Table 2-1 for definition of symbols.) This constraint insures

that the influence fields (detection radii) of the HVT and the decoys shall never overlap. If the decoys are capable of movement, the HVT is allowed to move in an unrestricted manner and the decoys are required to remain clear of the HVT. Obviously, if the decoys are considered to be stationary, i.e., VDC is very small, this condition must be reversed. If overlap should occur at the time of determination of initial positions, the initial positions of the decoys are adjusted so as to eliminate the overlap.

If the submarine searcher comes within detection range of a decoy, the submarine is considered to spend a period equivalent to TC hours classifying the decoy. During this time, the submarine is precluded from making new detections on either the HVT or other decoys. The period TC is also sometimes referred to as "decoy capture time" or decoy hold time."

During the period TC, the decoy continues to move in accordance with the appropriate pseudo-random procedures. The submarine searcher is assumed to move in consonance with the decoy. The position of the decoy at the end of the period TC is taken to be the re-start position of the submarine searcher.

To simulate submarine memory of classified decoy locations, the decoys are "turned off" by the model for a time TM following the classification period, TC. This permits the submarine to clear the immediate

area of the decoy. Obviously, the period TM should be made at least as long as R1R/VS when specifying this input value.

If it should occur that the submarine is within detection range of two or more decoys at any given instant, the model assumes that the submarine will investigate and classify the nearest decoy, ignoring the others. If ever the HVT is within range, the model assumes that it will be detected and classified by the submarine without regard for whatever decoys may also be present. As in all other cases, the instant of first detection of the HVT by the submarine terminates the specific replication.

A typical example of the motion generated for the various units within this submodel is presented in Figure 5-1. In this example, 4 decoys are present within the objective area (D1, D2, D3, and D4 in Figure 5-1) along with the HVT and submarine searcher.

An example of the type of results obtained from exercising this submodel is shown in Figure 5-2.

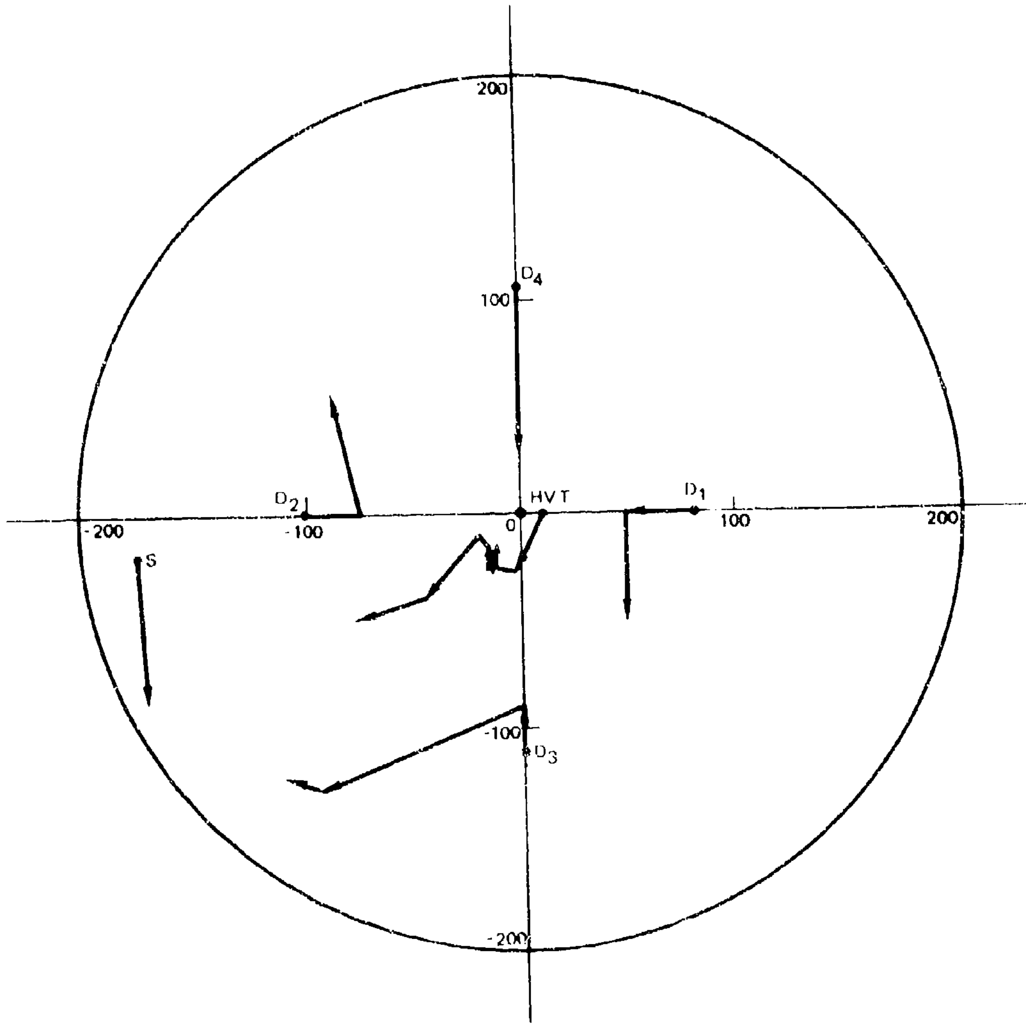

FIGURE 5-1 SAMPLE PLOT OF MOTIONS OF FOUR DECOYS (D_1, ... D_4), SUBMARINE, AND HVT FROM A TEST RUN WHERE ALL EXCEPT THE SUBMARINE ARE PREPOSITIONED INITIALLY

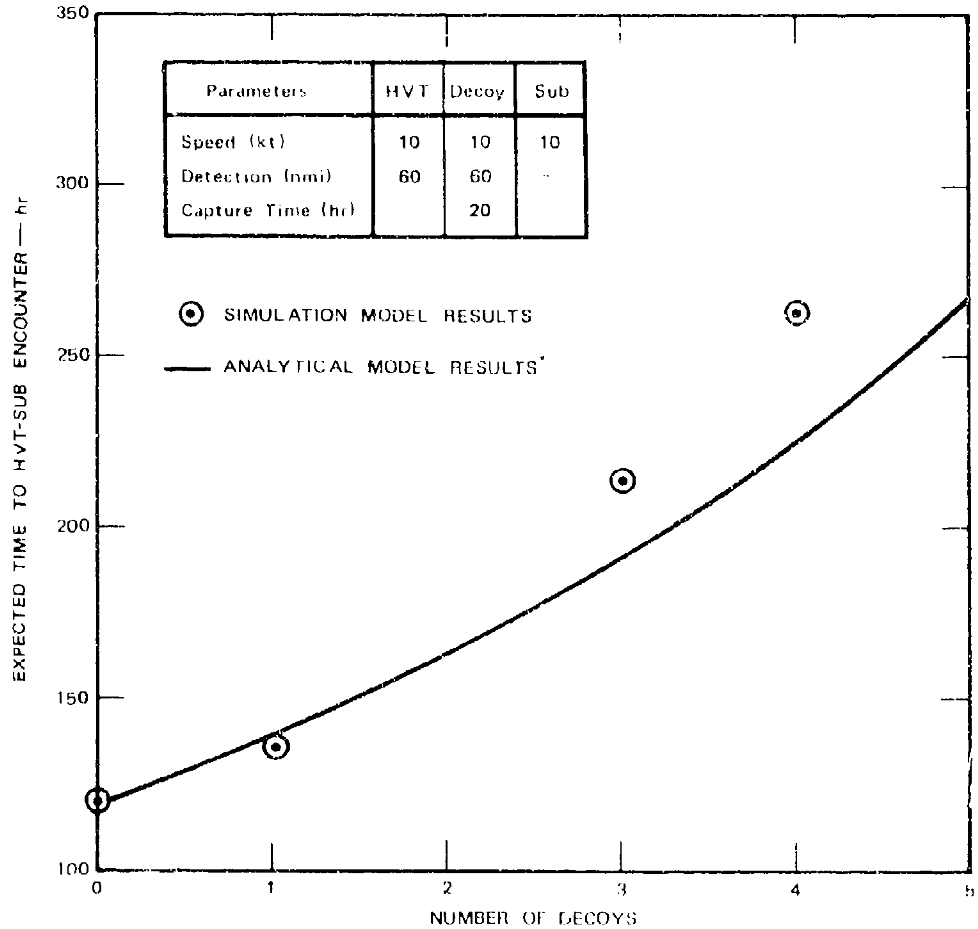

FIGURE 5-2 SAMPLE RESULTS FROM THE CONTINUOUS SEARCH — DECOY FIELD SUBMODEL

* J. M. Moore; "Semi-Markov Models of Search in the Presence of Decoys," NWRC RM-64, SRI Project 1016-245, Contract N00014-71-C-0119; Stanford Research Institute, Menlo Park, California; July 1971 (UNCLASSIFIED)

Appendix A

PROGRAM LISTING FOR CONTINUOUS SEARCH - NO DECOY SUBMODEL (MOD 1)

```
PROGRAM  MODI(INPUT,OUTPUT)
DIMENSION  TND(5),XD(5),YD(5),ANGD(5),IDN(5),COURD(5),
+ NOPD(5)
DIMENSION  VD(5)
DIMENSION  TDCVA(1000)
DIMENSION  TDCOY(100)
DATA  COURD/5.,5.,5.,5.,5./
DATA  VD/1.,1.,1.,1.,1./
R=200.
NINT=500
ITGT5=0
IP1=1
IP2=5
PRINT,*C*
PRINT,*CVA RANDOM START IN AREA*
PRINT,*SUB RANDOM START ON BOUNDARY*
PRINT,*C*
CALL  SECOND(X)
Y=RANF(X)
XX=0.
PRINT,*ENTER VC,VS,COURC,COURS,RO*
READ,VC,VS,COURC,COURS,RO
I=0
PRINT,*ENTER NREPLI*
READ,NREPLI
TIME=0.
HRS=0.
NDECOY=0
TDELAY=(-9.*VS)/50.+5.9
KOUNT=0
KOND=0
KONC=0
PRINT,*COUNTER,T ELASPED,T ENCOUNTERED*
202  CONTINUE
INC=0
INS=0
TNC=0.
TNS=0.
HRS=0.
DEG=57.+(17./60.)
TD=0.
NOPC=0
NOPS=0
DO  2  I=1,NDECOY
NOPD(I)=0
IDN(I)=0
TND(I)=0.
2    CONTINUE
200     CONTINUE
ANGC=6.28319*RANF(XX)
```

```
RADUS=R*RANF(XX)
XC=RADUS*COS(ANGC)
YC=RADUS*SIN(ANGC)
ANGS=6.28319*RANF(XX)
RAD=R
XS=RAD*COS(ANGS)
YS=RAD*SIN(ANGS)
IF(NDECOY.EQ.0)  GO  TO  800
DO  4  I=1,NDECOY
CALL  INDCOY(ANGD(I),XD(I),YD(I),XC,YC,XS,YS,XX,R,VC,VS)
TND(I)=TND(I)+1.
4  CONTINUE
800    CONTINUE
DT=0.5
15    CONTINUE
TIME=TIME+DT
HRS=HRS+DT
CALL LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,VC,COURC,XX,
NOPC,TDELAY,TD)
XC=XNOW
YC=YNOW
CALL LITBAL(TNS,XS,YS,ANGS,XNOW,YNOW,INS,VS,COURS,XX,
NOPS,TDELAY,TD)
XS=XNOW
YS=YNOW
IF(NOPS.EQ.2)  GO  TO  360
IF(NDECOY.EQ.0)  GO  TO  350
DO  350  I=1,NDECOY
CALL LITBAL(TND(I),XD(I),YD(I),ANGD(I),XNOW,YNOW,IDN(I),
VD(I),COURD(I),XX,NOPD(I),TDELAY,TD)
XD(I)=XNOW
YD(I)=YNOW
350    CONTINUE
CALL   MEET(VC,VS,XC,YC,XS,YS,IEN)
IF(IEN.EQ.1)  GO  TO  100
GO  TO  300
100  CONTINUE
KOUNT=KOUNT+1
KONC=KONC+1
TDCVA(KONC)=HRS
IPRINT=IP1*10
IF(IPRINT.NE.KOUNT)  GO  TO  1000
IP1=IP1+1
PRINT 95,KOUNT,HRS,TIME
95  FORMAT(I10,2F11.4/)
1000  CONTINUE
IF(KOUNT.EQ.NREPLI)  GO  TO  97
ISTAT=IP2*10
IF(ISTAT.NE.KOUNT)  GO  TO  1010
CALL   TIC(TDCVA,TDCOY,KONC,KOND,EXPT,VAR,ISTAT,VS)
```

```
CALL  FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
PRINT,*TIME GT 5000*
PRINT,ITGT5
IP2=IP2+5
GO  TO  202
1010    CONTINUE
GO  TO  202
300    CONTINUE
IF(NDECOY.EQ.0)  GO  TO  15
DO  361  I=1,NDECOY
CALL  MEET(VC,VS,XD(I),YD(I),XS,YS,IEN)
IF(IEN.EQ.1)  GO  TO  500
361  CONTINUE
IF(KOUNT.EQ.NREPLI)  GO  TO  97
GO  TO  15
310    CONTINUE
IF(NOPS.EQ.1)  GO  TO  15
PRINT,*MEET DECOY*
PRINT,I,XD(I),YD(I)
PRINT,XS,YS
IKEEP=I
NOPS=1
NOPD(I)=2
TD=0.
GO  TO  15
360  CONTINUE
NOPS=0
NOPD(IKEEP)=0
XS=XD(IKEEP)
YS=YD(IKEEP)
DO  370  I=1,NDECOY
IF(I.EQ.IKEEP)  GO  TO  370
CALL  MEET(VC,VS,XD(I),YD(I),XS,YS,IEN)
IF(IEN.EQ.1)  GO  TO  310
 370    CONTINUE
IKEEP=0
GO  TO  15
99    CONTINUE
500  PRINT ,*DECOY ENCOUNTERED,T ENCOUNTERED*
PRINT,I,HRS
98    CONTINUE
KOND=KOND+1
TDCOY(KOND)=HRS
KOUNT=KOUNT+1
GO  TO  202
97    CONTINUE
IF(KOND.EQ.0)  GO  TO  504
504    CONTINUE
CALL  TIC(TDCVA,TDCOY,KONC,KOND,EXPT,VAR,NREPLI,VS)
CALL  FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
```

MODI 06/10/71.

```
01600  PRINT,*FREQUENCE GT 5000*
01610  PRINT,ITGT5
01620  END
```

MODI 06/10/71.

```
SUBROUTINE  STAT(TIMF,ICOUT,FEAN,STDFV,VAR)
DIMENSION  TIME(1000)
SUM=0.
DO  5  I=1,ICOUT
SUM=SUM+TIMF(I)
5     CONTINUE
FEAN=SUM/FLOAT(ICOUT)
VAR=0.
DO  10  I=1,ICOUT
VAR=VAR+(TIMF(I)-FFAN)**2
10     CONTINUE
VAR=VAR/FLOAT(ICOUT)
STDEV=SQRT(VAR)
RETURN
END
```

```
SUBROUTINE TIC(TDCVA,TDCOY,KONC,KOND,EXPT,VAR,NREPLI,VS)
DIMENSION TDCVA(1000),TDCOY(100)
PRINT,*NO OF REPLICATIONS*
PRINT,NREPLI
TDLAY=(-9.*VS)/50.+5.9
IF(KOND.EQ.0) GO TO 10
CALL STAT(TDCOY,KOND,MEAND,STDEVD,VARD)
PRINT,MEAND,STDEVD,VARD
10 CONTINUE
PRINT,*MEANC,STDEVC,VARC*
CALL STAT(TDCVA,KONC,FEANC,STDEVC,VARC)
PRINT,FEANC,STDEVC,VARC
IF(KOND.EQ.0) RETURN
EXPT=(P/(1.-P))*(TDLAY+MEAND)+TDLAY+MEANC
TERM1=P/((1.-P)**2)
TERM2=(TDLAY+MEAND)**2
TERM3=(P/(1.-P))*VARD
PRINT,TERM1,TERM2,TERM3
VAR=TERM1*TERM2+TERM3+VARC
RETURN
END
```

```
SUBROUTINE  LITBAL(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,V,
 COURSE,XX,NOP,TDELAY,TD)
IF(NOP.EQ.1)  GO  TO  30
DT=.5
R=200.
IF(IND.EQ.0)  GO  TO  20
FLEN=TN*DT*V
IF(FLEN.GT.COURSE)  GO  TO  20
DX=(DT*V)*COS(ALAST)
DY=(DT*V)*SIN(ALAST)
RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
IF(RNOW.GT.R)  GO  TO  20
XNOW=XLAST+DX
YNOW=YLAST+DY
GO  TO  10
30  CONTINUE
XNOW=XLAST
YNOW=YLAST
IF(TD.LE.TDELAY)  GO  TO  15
PRINT,*LAST TD*
PRINT,TD
IEN=0
NOP=2
TD=0.
ALAST=6.28319*RANF(XX)
IND=1
TN=1.
PRINT,*T DELAY END*
GO  TO  15
15    TD=TD+DT
PRINT,*TD*
PRINT,TD
GO  TO  25
20    CONTINUE
IND=1
TN=1.
ALPHA=6.28319*RANF(XX)
PHY=ALPHA+ALAST
PHY=AMOD(PHY,6.28319)
DX=(DT*V)*COS(PHY)
DY=(DT*V)*SIN(PHY)
RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
IF(RNOW.GT.R)  GO  TO  20
ALAST=PHY
XNOW=XLAST+DX
YNOW=YLAST+DY
10    TN=TN+1.
25    CONTINUE
RETURN
END
```

MOD1 06/10/71.

```
SUBROUTINE INDCOY(ANGD1,XD1,YD1,XC,YC,XS,YS,XX,R,VC,VS)
5     CONTINUF
ANGD1=6.28319*RANF(XX)
XD1=R*COS(ANGD1)
YD1=R*SIN(ANGD1)
CALL  MEET(VC,VS,XD1,YD1,XS,YS,IEN)
IF(IEN.EQ.1)  GO  TO  5
RETURN
END
```

MOD1 06/10/71.

```
SUBROUTINE  MEET(VC,VS,XC,YC,XS,YS,IEN)
R0=60.
RANGE=SQRT((XC-XS)**2+(YC-YS)**2)
IF(RANGE.GT.R0)  GO  TO  10
IEN=1
GO  TO  20
10  IEN=0
20  CONTINUE
RETURN
END
```

```
SUBROUTINE  FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
DIMENSION  TDCVA(1000),IFRE(500),TCOM(500)
DIMENSION  IFREC(500)
ITGT5=0
IF(ISTAT.LT.NREPLI)  GO  TO  5
ISUM=NREPLI
GO  TO  6
5    ISUM=ISTAT
6    TINT=10.
TMAX=0.
DO  7   =1,NINT
IFRE(J)=0
IFREC(J)=0
7    CONTINUE
DO  10    I=1,ISUM
DO  20    J=1,NINT
TCOMPA=FLOAT(J)*TINT
IF(TDCVA(I).GT.5000.)  GO  TO  40
IF(TDCVA(I).GT.TCOMPA)  GO  TO  20
IFRE(J)=IFRE(J)+1
IF(TCOMPA.LT.TMAX)  GO  TO  10
TMAX=TCOMPA
GO  TO  10
40    ITGT5=ITGT5+1
GO  TO  10
20    CONTINUE
10    CONTINUE
IP=TMAX/TINT+1
DO  30  I=1,IP
TCOM(I)=FLOAT(I)*TINT
30    CONTINUE
IFREC(1)=IFRE(1)
DO  110  I=2,IP
IFREC(I)=IFRE(I)+IFREC(I-1)
110    CONTINUE
PRINT,*TIME INTERVAL,FREQUENCE*
DO  100  I=1,IP
IF(IFRE(I).EQ.0)  GO  TO  100
PRINT 35,TCOM(I),IFRE(I),IFREC(I)
35  FORMAT(1H ,2X,E11.2,2I10)
100    CONTINUE
RETURN
END
```

Appendix B

PROGRAM LISTING FOR SPRINT/DRIFT SEARCH -
NO DECOY SUBMODEL (SPRINT)

SPRINT 06/10/71.

```
PROGRAM  SPRINT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3)
CALL  RETR(1,5HTAPE1)
CALL  RETR(2,5HTAPE2)
REWIND 3
USE(LITBAL)
PRINT,*LAST FILE UPDATE?*
READ,LFILE
PRINT,*LAST REC CREATED?*
READ,LREC
PRINT,*VSSP,RDS,SPP?*
READ,VSSP,RDS,SPP
PRINT,*VSDF,RDD,DFP?*
READ,VSDF,RDD,DFP
PRINT,*SUB RANDOM START IN AREA=1,ON BOUNDARY=2,WHICH?*
READ,ISTART
R=200.
PRINT,*NO OF REPLICATIONS?*
READ,NREPLI
PRINT,*VC,COURC?*
READ,VC,COURC
XX=0.
PRINT,*ENTER IP*
READ,IP
ICOPY=IP*20
K=LFILE
KOUNT=LREC
CALL  SECOND(X)
Y=RANF(X)
220     CONTINUE
ANGC=6.28319*RANF(XX)
RADUS=R*RANF(XX)
XC=RADUS*COS(ANGC)
YC=RADUS*SIN(ANGC)
ANGS=6.28319*RANF(XX)
GO  TO  (10,20),ISTART
10  RADUS=R*RANF(XX)
GO  TO  25
20  RADUS=R
25  CONTINUE
XS=RADUS*COS(ANGS)
YS=RADUS*SIN(ANGS)
DT=.5
TIME=0.
INC=0
INS=0
TNC=0.
TNS=0.
Y=RANF(XX)
ISPSW=0
IF(Y.GT..5)  ISPSW=1
```

```
CALL  LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,VC,COURC,XX,
DT,R)
XC=XNOW
YC=YNOW
IF(ISPSW.EQ.1)  GO  TO  30
TMSP=TIME+DFP
VS=VSDF
COURS=VSDF*DFP
RD=RDD
GO  TO  35
30  TMSP=TIME+SPP
VS=VSSP
COURS=VSSP*SSP
RD=RDS
35  INS=0
TNS=0.
150  CONTINUE
CALL  MOVS(TNS,XS,YS,ANGS,XNOW,YNOW,INS,VS,COURS,XX,
DT,R)
XS=XNOW
YS=YNOW
DIST=SQRT((XC-XS)**2+(YC-YS)**2)
IF(DIST.LE.RD)  GO  TO  100
IF(TMSP.EQ.TIME)  GO  TO  110
IF(TMSP.GT.TIME)  GO  TO  120
PRINT,*ERR1-TMSP.LE.TIME*
110    CONTINUE
IF(ISPSW.EQ.1)  GO  TO  130
ISPSW=1
TMSP=TIME+SPP
VS=VSSP
RD=RDS
GO  TO  140
130  CONTINUE
ISPSW=0
TMSP=TIME+DFP
VS=VSDF
COURS=VSDF*DFP
RD=RDD
140  INS=0
TNS=0.
12   TIME=TIME+DT
CALL  LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,VC,COURC,XX,
DT,R)
XC=XNOW
YC=YNOW
GO  TO  150
100  CONTINUE
KOUNT=KOUNT+1
WRITE(3,160)  KOUNT,TIME
```

```
160  FORMAT(I10,E20.4)
IF(KOUNT.LT.ICOPY)  GO  TO  220
IF(IREC.NE.0)  GO  TO  404
REWIND 1
REWIND 3
WRITE(1,413) KOUNT
407  READ(3,160)  I,TIME
IF(I.EQ.KOUNT)  GO  TO  406
WRITE(1,160)  I,TIME
GO  TO  407
406  WRITE(1,160)  I,TIME
ENDFILE 1
REWIND 1
CALL  REPL(1,5HTAPE1)
REWIND 3
IREC=KOUNT
K=1
GO  TO  310
404  CONTINUE
GO  TO  (400,410) K
400  CONTINUE
REWIND 1
REWIND  2
REWIND  3
WRITE(2,413)  KOUNT
READ(1,413)  NREC
413  FORMAT(I10)
IREC=0
IF(NREC.EQ.0)  GO  TO  600
415  READ(1,160)  I,TIME
IF(I.EQ.NREC)  GO  TO  411
IREC=IREC+1
WRITE(2,160)  IREC,TIME
GO  TO  415
411  IREC=IREC+1
WRITE(2,160)  IREC,TIME
600  READ(3,160)  I,TIME
IF(IREC.EQ.KOUNT)  GO  TO  420
IF(I.EQ.KOUNT)  GO  TO  420
IREC=IREC+1
WRITE(2,160)  IREC,TIME
GO  TO  600
420  IREC=IREC+1
WRITE(2,160)  IREC,TIME
ENDFILE 2
REWIND 2
CALL  REPL(2,5HTAPE2)
K=2
REWIND 1
REWIND 2
```

```
REWIND 3
GO  TO  310
410  CONTINUE
REWIND 1
REWIND 2
REWIND 3
WRITE(1,413)  KOUNT
READ(2,413)  NREC
IREC=0
IF(NREC.EQ.0)  GO  TO  605
610  READ(2,160)  I,TIME
IF(I.EQ.NREC)  GO  TO  615
IREC=IREC+1
WRITE(1,160)  IREC,TIME
GO  TO  610
615  IREC=IREC+1
WRITE(1,160)  IREC,TIME
605  READ(3,160)  I,TIME
IF(IREC.EQ.KOUNT)  GO  TO  620
IF(I.EQ.KOUNT)  GO  TO  620
IREC=IREC+1
WRITE(1,160)  IREC,TIME
GO  TO  605
620  IREC=IREC+1
WRITE(1,160)  IREC,TIME
ENDFILE 1
REWIND 1
CALL  REFL(1,SHTAPE1)
K=1
REWIND 1
REWIND 2
REWIND 3
GO  TO  310
310  PRINT,*LAST REC CREATED,LAST FILE CREATED*
PRINT,KOUNT,K
IF(KOUNT.EQ.NREPLI)  GO  TO  300
IP=IP+1
PRINT,*NEXT IP IS*
PRINT,IP
ICOPY=IP*20
GO  TO  220
300  CONTINUE
IP=IP+1
PRINT,*NEXT IP IS*
PRINT,IP
END
```

SPRINT 06/10/71.

```
SUBROUTINE MOVS(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,V,
+ COURSE,XX,DT,R)
IF(IND.EQ.0) GO TO 20
DX=DT*V*COS(ALAST)
DY=DT*V*SIN(ALAST)
RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
IF(RNOW.GT.R) GO TO 20
XNOW=XLAST+DX
YNOW=YLAST+DY
GO TO 10
20 CONTINUE
IND=1
ALPHA=6.28319*RANF(XX)
PHY=ALPHA+ALAST
PHY=AMOD(PHY,6.28319)
DX=DT*V*COS(PHY)
DY=DT*V*SIN(PHY)
RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
IF(RNOW.GT.R) GO TO 20
ALAST=PHY
XNOW=XLAST+DX
YNOW=YLAST+DY
10 CONTINUE
TN=TN+1.
RETURN
END
```

LITBAL   06/10/71.

```
      SUBROUTINE  LITBAL(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,V,COURSE,XX
      IF(IND.EQ.0)  GO  TO  20
      FLEN=TN*DT*V
      IF(FLEN.GT.COURSE)  GO  TO  20
      DX=(DT*V)*COS(ALAST)
      DY=(DT*V)*SIN(ALAST)
      RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
      IF(RNOW.GT.R)  GO  TO  20
      XNOW=XLAST+DX
      YNOW=YLAST+DY
      GO  TO  10
 20   CONTINUE
      IND=1
      TN=1.
      ALPHA=6.28319*RANF(XX)
      PHY=ALPHA+ALAST
      PHY=AMOD(PHY,6.28319)
      DX=(DT*V)*COS(PHY)
      DY=(DT*V)*SIN(PHY)
      RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
      IF(RNOW.GT.R)  GO  TO  20
      ALAST=PHY
      XNOW=XLAST+DX
      YNOW=YLAST+DY
 10   TN=TN+1.
      RETURN
      END
```

```
PROGRAM STATICS(INPUT,OUTPUT,TAPE4,TAPE5,
TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12)
13  CONTINUE
PRINT,*ENTER FILE NO*
READ,K
GO  TO  (3,3,3,4,5,6,7,8,9,10,11,12),K
3  CONTINUE
PRINT,*ERR-FILE NOT USE ,TRY AGAIN*
GO  TO  13
4  NAME=5HTAPE4
GO  TO  14
5  NAME=5HTAPE5
GO  TO  14
6  NAME=5HTAPE6
GO  TO  14
7  NAME=5HTAPE7
GO  TO  14
8  NAME=5HTAPE8
GO  TO  14
9  NAME=5HTAPE9
GO  TO  14
10  NAME=6HTAPE10
GO  TO  14
11  NAME=6HTAPE11
GO  TO  14
12  NAME=6HTAPE12
GO  TO  14
14  CONTINUE
CALL  RETR(K,NAME)
REWIND K
READ(K,55)KOUNT
55  FORMAT(I10)
IF(KOUNT.NE.0)  GO  TO  20
PRINT,*0 REC ON FILE K,TRY AGAIN*
GO  TO  13
20  CONTINUE
SUM=0.
50  READ(K,30)  I,TIME
30  FORMAT(I10,E20.4)
SUM=SUM+TIME
IF(I.NE.KOUNT)  GO  TO  50
ISUM=I
IF(ISUM.EQ.KOUNT)  GO  TO  60
PRINT,*ERROR-KOUNT NOT MATCH LAST REC NO*
60  PRINT,*ISUM,SUM*
PRINT,ISUM,SUM
FEAN=SUM/FLOAT(ISUM)
REWIND K
SUM=0.
80  READ(K,30)  I,TIME
```

```
X=(TIME-FEAN)**2
SUM=SUM+X
IF(I.NE.KOUNT)  GO  TO  80
GO  TO  70
70  VAR=SUM/F )AT(ISUM)
STAN=SQRT(VAR
PRINT,*SAMPLE SIZE IS*
PRINT,ISUM
PRINT,*MEAN,STANDARD DEVIATION,VARIANCE*
PRINT,FEAN,STAN,VAR
RETURN
END
```

```
PROGRAM  UPDATE(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,
TAPE4,TAPE5,TAPE6,TAPE7,TAPE8,TAPE9)
2  CONTINUE
PRINT,*THIS IS DATA FILE UPDATE*
PRINT,*ENTER FILE NO*
READ,IFILE
PRINT,*ENTER,K AND KOUNT, K=1 OR 2 FOR FILE TAPE1 OR TAPE2*
PRINT,*KOUNT=NUMBER OF RECORDS TO BE INSERTED*
READ,K,KOUNT
CALL  RETR(1,5HTAPE1)
CALL  RETR(2,5HTAPE2)
GO  TO  (1,1,1,4,5,6,7),IFILE
1  PRINT,*ERROR-IFILE LESS THAN 4,TRY AGAIN*
PRINT,IFILE
GO  TO  2
4  NAME=5HTAPE4
GO  TO  88
5  NAME=5HTAPE5
GO  TO  88
6  NAME=5HTAPE6
GO  TO  88
7  NAME=5HTAPE7
GO  TO  88
8  NAME=5HTAPE8
GO  TO  88
9  NAME=5HTAPE9
GO  TO  88
88  CALL  RETR(IFILE,NAME)
REWIND IFILE
REWIND 1
REWIND 2
REWIND 3
120  CONTINUE
READ(IFILE,101) ILAST
PRINT,ILAST
IF(ILAST.EQ.0)  GO  TO  888
101  FORMAT(I10)
NREC=KOUNT+ILAST
WRITE(3,101)  NREC
PRINT,NREC
ICOUN=1
130  CONTINUE
READ(IFILE,100) I,TIME
100  FORMAT(I10,E20.4)
IF(ICOUN.EQ.ILAST)  GO  TO  110
WRITE(3,100)  I,TIME
ICOUN=ICOUN+1
GO  TO  130
110  WRITE(3,100) ICOUN,TIME
ICOUN=ILAST+1
```

```
150   REWIND K
READ(K,101) I
151   CONTINUE
READ(K,100)  I,TIME
WRITE(3,100) ICOUN,TIME
IF(ICOUN.EQ.NREC)  GO  TO  140
ICOUN=ICOUN+1
GO  TO  151
140   WRITE(3,100) ICOUN,TIME
ENDFILE  3
REWIND 3
REWIND IFILE
READ(3,101)  NREC
WRITE(IFILE,101) NREC
240   READ(3,100) ICOUN,TIME
IF(ICOUN.EQ.NREC)   GO  TO  230
WRITE(IFILE,100)  ICOUN,TIME
GO  TO  240
230   WRITE(IFILE,100) ICOUN,TIME
ENDFILE IFILE
REWIND IFILE
CALL  REPL(IFILE,NAME)
PRINT,*FILE UPDATE IS,LAST REC CREATED IS*
PRINT,IFILE,ICOUN
GO  TO  99
888   CONTINUE
REWIND IFILE
NREC=KOUNT
WRITE(IFILE,101)  NREC
READ(K,725) KOUNT
725   FORMAT(I10)
ICOUN=1
210   CONTINUE
READ(K,100)  I,TIME
IF(I.EQ.KOUNT)  GO  TO  200
IF(ICOUN.EQ.KOUNT)  GO  TO  200
WRITE(IFILE,100)  ICOUN,TIME
ICOUN=ICOUN+1
GO  TO  210
200   WRITE(IFILE,100)  ICOUN,TIME
PRINT,*FILE UPDATE IS,LAST REC CREATED*
PRINT,IFILE,ICOUN
ENDFILE IFILE
REWIND IFILE
CALL  REPL(IFILE,NAME)
99   CONTINUE
END
```

FILECR   06/10/71.

```
PROGRAM   FILECR(INPUT,OUTPUT,TAPE1,TAPE2,
TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12)
DO   100   I=4,12
GO   TO   (4,4,4,4,5,6,7,8,9,10,11,12),I
4  NAME=5HTAPE1
IFILE=1
GO   TO   110
5  NAME=5HTAPE2
IFILE=2
GO   TO   110
6  NAME=5HTAPE6
IFILE=6
GO   TO   110
7  NAME=5HTAPE7
IFILE=7
GO   TO   110
8  NAME=5HTAPE8
IFILE=8
GO   TO   110
9  NAME=5HTAPE9
IFILE=9
GO   TO   110
10  NAME=6HTAPE10
IFILE=10
GO   TO   110
11  NAME=6HTAPE11
IFILE=11
GO   TO   110
12  NAME=6HTAPE12
IFILE=12
110   CONTINUE
REWIND IFILE
ICOUN=0
II=1
TIME=0.
WRITE(IFILE,30) ICOUN
30   FORMAT(I10)
WRITE(IFILE,35)   II,TIME
35   FORMAT(I10,E20.4)
ENDFILE   IFILE
REWIND IFILE
100   CONTINUE
END
```

Appendix C

PROGRAM LISTING FOR CONTINUOUS SEARCH -
DECOY FIELD SUBMODEL (MOD 3)

```
PROGRAM  MOD3(INPUT,OUTPUT,TAPE1,TAPE2)
DIMENSION  TND(5),XD(5),YD(5),ANGD(5),IND(5),COURD(5)
DIMENSION  NOPD(5),ICHASD(5),VDC(5)
DIMENSION  TDCVA(500),TDCOY(500),KPD(500)
DIMENSION  VD(5)
DIMENSION  DTMON(5)
DIMENSION  IPAR(5)
PRINT,*ENTER NO OF DECOYS*
READ,NDECOY
DO  2000  NI=1,NDECOY
PRINT 2010,NI
2010  FORMAT(1H ,*ENTER FOR DECOY NO*,I3,2X,*COURD, VD*)
READ,COURD(NI),VD(NI)
2000  CONTINUE
PRINT,*ENTER VC,VS,COURC,COURS*
READ,VC,VS,COURC,COURS
PRINT,*ENTER TC,TM,R1R,R2R*
READ,TC,TM,R1R,R2R
PRINT,*ENTER NO OF REPLICATIONS(NOT TO EXCEED 500)*
READ,NREPLI
NINT=500
ITGT5=0
IP1=1
IP2=5
R=200.
PRINT,*C*
PRINT,*CVA,DECOYS RANDOM START IN AREA*
PRINT,*SUB RANDOM START ON BOUNDARY*
PRINT,*C*
PRINT,*COUNTER   TIME ENCOUNTERED*
CALL  SECOND(X)
Y=RANF(X)
XX=0.
TIME=0.
HRS=0.
KOUNT=0
KOND=0
KONC=0
15  CONTINUE
DO  662  IDY=1,NDECOY
DTMON(IDY)=0.
662   CONTINUE
TIME=0.
HRS=0.
INC=0
INS=0
TNC=0.
TNS=0.
ICHASS=0
DEG=57.+(17./60.)
```

```
DO 2 I=1,NDECOY
IND(I)=0
TND(I)=0.
ICHASD(I)=0
2 CONTINUE
200 CONTINUE
ANGC=6.28319*RANF(XX)
RADUS=R*RANF(XX)
XC=RADUS*COS(ANGC)
YC=RADUS*SIN(ANGC)
TNC=TNC+1.
ANGS=6.28319*RANF(XX)
RAD=R
XS=RAD*COS(ANGS)
YS=RAD*SIN(ANGS)
TNS=TNS+1.
INC=1
INS=1
IF(NDECOY.EQ.0) GO TO 800
DO 4 I=1,NDECOY
CALL INDCOY(ANGD(I),XD(I),YD(I),XC,YC,XS,YS,
+ XX,R,VDC(I),VS,VC)
TND(I)=TND(I)+1.
IND(I)=1
4 CONTINUE
800 CONTINUE
DT=0.5
202 CONTINUE
TIME=TIME+DT
HRS=HRS+DT
IC=0
CALL LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,
+ VC,COURC,XX,VC,VS,VDC,XD,YD,IC,NDECOY,
+XC,YC)
XC=XNOW
YC=YNOW
IF(ICHASS.EQ.0) GO TO 300
IF(STMON.NE.TIME) GO TO 305
DO 310 I=1,NDECOY
IF(ICHASD(I).EQ.1) GO TO 315
310 CONTINUE
PRINT,*ERR-ENCOUNTERED DECOY NOT TURNED ON*
PRINT 3,I
3 FORMAT(I10)
GO TO 00202
315 XS=XD(I)
YS=YD(I)
TNS=1.
ANGS=6.28319*RANF(XX)
INS=1
```

```
ICHASS=0
GO TO 00320
300 CONTINUE
IC=88
CALL LITBAL(TNS,XS,YS,ANGS,XNOW,YNOW,INS,VS,
+ COURS,XX,VC,VS,VDC,XD,YD,IC,NDECOY,
+XC,YC)
XS=XNOW
YS=YNOW
GO TO 00320
305 IF(TIME.LT.STMON) GO TO 320
GO TO 00300
320 CONTINUE
IF(NDECOY.EQ.0) GO TO 370
DO 330 I=1,NDECOY
IF(ICHASD(I).EQ.1) GO TO 331
IF(TIME.LT.DTMON(I)) GO TO 330
IF(TIME.EQ.DTMON(I)) GO TO 330
350 CONTINUE
IC=I
CALL LITBAL(TND(I),XD(I),YD(I),ANGD(I),XNOW,YNOW,
+ IND(I),VD(I),COURD(I),XX,VC,VS,VDC,XD,YD,
+IC,NDECOY,XC,YC)
XD(I)=XNOW
YD(I)=YNOW
GO TO 00330
331 CONTINUE
IF(TIME.LT.DTMON(I)) GO TO 350
IF(TIME.EQ.DTMON(I)) GO TO 345
IF((TIME-DTMON(I)).LT.DT) GO TO 342
PRINT,*ERR-ICHASD=1,DTMON.LT.TIME*
PRINT,TIME,DTMON(I),I
ICHASD(I)=0
GO TO 00330
342 CONTINUE
ICHASD(I)=0
ICHASS=0
DTMON(I)=DTMON(I)+TM
XS=XD(I)
YS=YD(I)
INS=1
TNS=2.
ANGS=6.28319*RANF(XX)
GO TO 00202
345 CONTINUF
ICHASD(I)=0
DTMON(I)=DTMON(I)+TM
ICHASS=0
XS=XD(I)
YS=YD(I)
```

```
ANGS=6.28319*RANF(XX)
INS=1
TNS=2.
GO  TO 00330
330     CONTINUE
370  CONTINUE
420     CONTINUE
CALL  MEET(VC,VS,XC,YC,XS,YS,IFN)
IF(IFN.EQ.1)  GO  TO  100
GO  TO 00390
390     CONTINUE
IF(ICHASS.EQ.1)  GO  TO  202
IF(NDECOY.EQ.0)  GO  TO  202
ISAVE=0
DO  440  I=1,NDECOY
IF(ICHASD(I).EQ.1)  GO  TO  202
IF(TIME.LE.DTMON(I))  GO  TO  440
CALL  MEET(VDC(I),VS,XD(I),YD(I),XS,YS,IFN)
IF(IFN.EQ.0)  GO  TO  440
DISM=SQRT((XD(I)-XS)**2+(YD(I)-YS)**2)
ISAVE=I
440     CONTINUE
IF(ISAVE.EQ.0)  GO  TO  202
DO  450  I=1,NDECOY
IF(I.EQ.ISAVE)  GO  TO  450
IF(TIME.LE.DTMON(I))  GO  TO  450
CALL  MEET(VDC(I),VS,XD(I),YD(I),XS,YS,IEN)
IF(IEN.EQ.0)  GO  TO  450
DIS=SQRT((XD(I)-XS)**2+(YD(I)-YS)**2)
IF(DIS.GT.DISM)  GO  TO  450
DISM=DIS
ISAVE=I
450     CONTINUE
DTMON(ISAVE)=TIME+TC
ICHASD(ISAVE)=1
STMON=TIME+TC
ICHASS=1
GO  TO 00460
100     KOUNT=KOUNT+1
KONC=KONC+1
TDCVA(KONC)=HRS
IPRINT=IP1*1
IF(IPRINT.NE.KOUNT)  GO  TO  1000
IP1=IP1+1
PRINT 95,KOUNT,TDCVA(KONC)
95  FORMAT(I10,E11.4)
WRITE(1,95)  KOUNT,TDCVA(KONC),TIME
1000    CONTINUE
IF(KOUNT.EQ.NREFLI)  GO  TO  97
ISTAT=IP2*10
```

```
IF(ISTAT.NF.KOUNT)   GO   TO   1010
CALL   STAT(TDCVA,KONC,FEANC,STDFVC,VARC)
PRINT,*MEANC,STDFVC,VARC*
PRINT,FEANC,STDEVC,VARC
CALL   FREQ(TDCVA,KONC,ISTAT.NRFPLI,NINT,ITGT5)
PRINT,*FREQUENCY GT 5000.-CVA*
PRINT,ITGT5
IF(KOND.EQ.0)   GO   TO   15
CALL   FREQ(TDCOY,KOND,KOND,KOND,NINT,ITGT5)
PRINT,*FRFQUFNCY GT 5000.-DECOY*
PRINT,ITGT5
IP2=IP2+5
GO   TO   15
460     CONTINUF
KOND=KOND+1
TDCOY(KOND)=HRS
KPD(KOND)=KOUNT+1
GO   TO 00202
1010   CONTINUE
GO   TO   15
97     CONTINUE
CALL  STAT(TDCVA,KONC,FEANC,STDEVC,VARC)
PRINT,*MEANC,STDFVC,VARC*
PRINT176,FEANC,STDEVC,VARC
176     FORMAT(3F11.4)
CALL   FREQ(TDCVA,KONC,ISTAT,NRFPLI,NINT,ITGT5)
PRINT,*FREQUENCY GT 5000.-CVA*
PRINT,ITGT5
GO   TO   999
IF(KOND.EQ.0)   GO   TO   999
CALL   FREQ(TDCOY,KOND,KOND,KOND,NINT,ITGT5)
PRINT,*FREQUFNCY GT 5000.-DECOY*
PRINT,ITGT5
IF(KOND.EQ.0)   GO   TO   999
CALL   STAT(TDCOY,KOND,FEAND,STDEVD,VARD)
PRINT,*MEAND,STDEVD,VARD*
PRINT176,FEAND,STDEVD,VARD
999     CONTINUF
111     CONTINUF
END
```

MOD3 06/11/71.

```
SUBROUTINE STAT(TIME,ICOUT,FEAN,STDEV,VAR)
DIMENSION  TIME(500)
SUM=0.
DO  5  I=1,ICOUT
SUM=SUM+TIME(I)
5     CONTINUE
FEAN=SUM/FLOAT(ICOUT)
VAR=0.
DO  10  I=1,ICOUT
VAR=VAR+(TIME(I)-FEAN)**2
10     CONTINUE
VAR=VAR/FLOAT(ICOUT)
STDEV=SQRT(VAR)
RETURN
END
```

```
SUBROUTINE  LITRAL(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,
V,COURSE,XX,VC,VS,VDC,XD,YD,IC,NDECOY,
XC,YC)
DIMENSION DYMR(5),SUMR(5)
DIMENSION  XD(5),YD(5),VDC(5)
DT=.5
R=200.
IF(IC.EQ.0)  GO  TO  20
IF(IC.EQ.88)  GO  TO  30
IF(IC.LE.NDECOY)  GO  TO  25
PRINT,*ERR,IC INCORRECT*
PRINT,IC
RETURN
20     CONTINUE
IF(IND.EQ.0)  GO  TO  15
CALL XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,
XNOW,YNOW,IND,ANOW,IANG)
45     DO  40  I=1,NDECOY
DYMR(I)=SQRT((XD(I)-XNOW)**2+(YD(I)-YNOW)**2)
RCS=R2R
VP=VDC(I)
SUMR(I)=RCS+R1R
IF(DYMR(I).LE.SUMR(I))  GO  TO  35
40     CONTINUE
IF(IANG.EQ.0)  GO  TO  999
ALAST=ANOW
IANG=0
GO  TO  999
35     IND=0
15  CALL NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,
XNOW,YNOW,IANG)
TN=2.
IND=1
IANG=1
GO  TO  45
25     CONTINUE
IF(IND.EQ.0)  GO  TO  50
CALL  XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,
XNOW,YNOW,IND,ANOW,IANG)
RCS=R2R
60     DYMR(IC)=SQRT((XC-XNOW)**2+(YC-YNOW)**2)
VP=VDC(IC)
SUMR(IC)=RCS+R1R
IF(DYMR(IC).LE.SUMR(IC))  GO  TO  55
IF(IANG.EQ.0)  GO  TO  999
ALAST=ANOW
IANG=0
GO  TO  999
55     CONTINUE
IND=0
```

```
50      CONTINUE
CALL    NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,
XNOW,YNOW,INAG)
IND=1
TN=2.
IANG=1
GO   TO   60
30      CONTINUE
IF(IND.EQ.0)   GO   TO   65
CALL    XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,XNOW,
YNOW,IND,ANOW,IANG)
IF(IANG.EQ.0)   GO   TO   999
ALAST=ANOW
IANG=0
GO   TO   999
65      CONTINUE
CALL    NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,
XNOW,YNOW,IANG)
TN=2.
IND=1
ALAST=ANOW
IANG=0
999     CONTINUE
RETURN
END
```

```
SUBROUTINE  MFET(VC,VS,XC,YC,XS,YS,IEN)
R0=60.
RANGE=SQRT((XC-XS)**2+(YC-YS)**2)
IF(RANGE.GT.R0)  GO  TO  10
IFN=1
GO  TO 00020
10  IFN=0
20  CONTINUE
RETURN
END
```

```
SUBROUTINE  INDCOY(ANGD1,XD1,YD1,XC,YC,XS,YS,XX,R,
VDC,VS,VC)
5     CONTINUE
ANGD1=6.28319*RANF(XX)
RAD=R*RANF(XX)
XD1=RAD*COS(ANGD1)
YD1=RAD*SIN(ANGD1)
CALL  MEET(VDC,VS,XD1,YD1,XS,YS,IEN)
IF(IEN.EQ.1)  GO  TO  5
RDS=R1R
RCS=R2R
SUMR=RDS+RCS
RCVADY=SQRT((XC-XD1)**2+(YC-YD1)**2)
IF(RCVADY.GT.SUMR)  RETURN
GO  TO  5
RETURN
END
```

MOD3 06/11/71.

```
      SUBROUTINE   XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,
     +XNOW,YNOW,IND,ANOW,IANG)
    5     CONTINUE
      FLEN=TN*DT*V
      IF(FLEN.GT.COURSE)   GO   TO   10
      DX=(DT*V)*COS(ALAST)
      DY=(DT*V)*SIN(ALAST)
      RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
      IF(RNOW.GT.R)   GO   TO   10
      XNOW=XLAST+DX
      YNOW=YLAST+DY
      TN=TN+1.
      IANG=0
      GO   TO   99
   10   CONTINUE
      CALL   NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,XNOW,YNOW,IANG)
      TN=2.
      IND=1
      IANG=1
   99     CONTINUE
      RETURN
      END
```

```
SUBROUTINE  NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,
ANOW,XNOW,YNOW,IANG)
20     CONTINUE
ALPHA=6.28319*RANF(XX)
PHY=ALPHA+ALAST
PHY=AMOD(PHY,6.28319)
DX=(DT*V)*COS(PHY)
DY=(DT*V)*SIN(PHY)
RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
IF(RNOW.GT.R)  GO  TO  20
ANOW=PHY
XNOW=XLAST+DX
YNOW=YLAST+DY
RETURN
END
```

```
SUBROUTINE  FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
DIMENSION  TDCVA(500),IFRE(500),TCOM(500)
DIMENSION  IFREC(500)
ITGT5=0
IF(ISTAT.LT.NREPLI)  GO  TO  5
ISUM=NREPLI
GO  TO  6
5    ISUM=ISTAT
6  TINT=10.
TMAX=0.
DO  7  J=1,NINT
IFRE(J)=0
IFREC(J)=0
7    CONTINUE
DO  10  I=1,ISUM
DO  20  J=1,NINT
TCOMPA=FLOAT(J)*TINT
IF(TDCVA(I).GT.5000.)  GO  TO  40
IF(TDCVA(I).GT.TCOMPA)  GO  TO  20
IFRE(J)=IFRE(J)+1
IF(TCOMPA.LT.TMAX)  GO  TO  10
TMAX=TCOMPA
GO  TO  10
40  ITGT5=ITGT5+1
GO  TO  10
20    CONTINUE
10    CONTINUE
IP=TMAX/TINT+1.
DO  30  I=1,IP
TCOM(I)=FLOAT(I)*TINT
30    CONTINUE
IFREC(1)=IFRE(1)
DO  110  I=2,IP
IFREC(I)=IFRE(I)+IFREC(I-1)
110    CONTINUE
PRINT,*TIME INTERVAL,FREQUENCE,CUM F*
DO  100  I=1,IP
IF(IFRE(I).EQ.0)  GO  TO  100
PRINT 35,TCOM(I),IFRE(I),IFREC(I)
35  FORMAT(1H ,2X,E11.2,2X,2I10)
100    CONTINUE
RETURN
END
```

DISTRIBUTION LIST

	No. of Copies
Director of Defense Research & Engineering Department of Defense Washington, D. C. 20301 (Tactical Systems Plans and Analysis)	1
Advanced Research Projects Agency Department of Defense Washington, D.C. 20301 (Technical Library)	1
Chief of Naval Operations Department of the Navy Washington, D.C. 20350	
(Op-952)	1
(Op-981)	1
(Op-96, SEAMIX Project Officer)	1
Office of Naval Research Department of the Navy Arlington, Virginia 22217	
(Code 462)	2
(Code 434)	1
Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101	1
Headquarters, Marine Corps Washington, D.C. 20380 (Dr. A.L. Slafkosky, Science Advisor)	1
Naval Air Systems Command Department of the Navy Washington, D.C. 20360	
(AIR-370)	1
(AIR- 3362)	1

	No. of Copies
Naval Ship Systems Command Department of the Navy Washington, D.C. 20360 (SHIP-902)	1
Commander Antisubmarine Warfare Force U.S. Pacific Fleet FPO San Francisco, California 96610 (CDR Fishburn)	1
Naval Weapons Laboratory (Code KCU) Dahlgren, Virginia 22448	1
Naval Ordnance Laboratory Silver Spring, Maryland 20910 (Technical Library)	1
Naval Academy Annapolis, Maryland 21402 (Technical Library)	1
Naval Postgraduate School Monterey, California 93940 (Department of Operations Research and Administrative Sciences)	2
Naval War College Newport, Rhode Island 02844 (Technical Library)	1
Naval Research Laboratory Washington, D.C. 20390 (Code 2620)	2
(Code 2629/ONR-L)	2
Naval Air Development Center Warminster, Pennsylvania 18974 (Mr. Madera)	1
Naval Ship Research and Development Center (Code OH74) Washington, D.C. 20034	1

	No. of Copies
Naval Undersea Research and Development Center San Diego, California 92152	1
New London Laboratory Naval Underwater Systems Center New London, Connecticut 06320	
(Mr. C. King)	1
Applied Physics Laboratory Johns Hopkins University 8621 Georgia Avenue Silver Spring, Maryland 20910	1
Center for Naval Analysis 1401 Wilson Boulevard Arlington, Virginia 22209	
(Library)	1
(Dr. S. Shear)	1
Institute of Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1
Arthur D. Little, Inc. Acorn Park Cambridge, Massachusetts 02139	
(Dr. J. M. Dobbie)	1
Ordnance Research Laboratory State College, Pennsylvania 16801	
(Dr. Janetta)	1
The Rand Corporation 1700 Main Street Santa Monica, California 90401	1
Santa Barbara Analysis and Planning Corp. P.O. Box 3798 Santa Barbara, California 93105	1
Ultrasystems, Inc. 500 Newport Center Drive Newport Beach, California 92660	1

	No. of Copies
Daniel H. Wagner Associates, Inc. Station Square One Paoli, Pennsylvania 19301	1
Presearch Incorporated 8720 Georgia Avenue Silver Spring, Maryland 20910	1